CSCI3656: NUMERICAL COMPUTATION
Homework 8: Due Friday, Oct.29, 5:00pm

Turn in your own writeup that includes your code. List any resources you used including collaborating with others. Submit a PDF on Canvas by Friday, Oct. 29 at 5pm.

The purpose of this homework is to see the relationship between the condition number and the numerical error when solving linear least-squares problems.

First, implement the following methods for least-squares, which you'll use in the next exercise.

1. Method of Normal Equations (uses the Cholesky factorization)
2. Method based on the Thin QR factorization

(Still not sure what these are after watching lectures? Ask on Piazza.) Next, load the given matrix into memory. Call the matrix A,

$$A = \begin{bmatrix} a_1 & \cdots & a_n \end{bmatrix}, \tag{1}$$

where $a_i \in \mathbb{R}^m$ is the ith column of A. Define the matrices $A_1, \ldots, A_n$ as:

$$A_k = \begin{bmatrix} a_1 & \cdots & a_k \end{bmatrix}, \quad k = 1, \ldots, n. \tag{2}$$

That is, A_k contains the first k columns of the matrix A (that you loaded into memory).

Now, generate the error data that you'll analyze. For k from $k_{\min} = 40$ to $k_{\max} = 65$:

1. Report the size, rank, and condition number of A_k.
2. Generate 100 random vectors $b_i \in \mathbb{R}^m$. For each b_i,
   (a) Use the built-in equation solver[1] to compute the least-squares minimizers given A_k and b_i. Call this vector x_{true}, because we're just gonna trust the software on this one.
   (b) Use your Normal Equation solver to compute the least-squares minimizer, x_{NE}. Compute the relative error with x_{true}:
   $$\text{err}_{k,i}^{\text{NE}} = \frac{\|x_{\text{NE}} - x_{\text{true}}\|_2}{\|x_{\text{true}}\|_2} \tag{3}$$
   (c) Use your QR solver to compute the least-squares minimizer, x_{QR}. Compute the relative error with x_{true}:
   $$\text{err}_{k,i}^{\text{QR}} = \frac{\|x_{\text{QR}} - x_{\text{true}}\|_2}{\|x_{\text{true}}\|_2} \tag{4}$$
3. For each of QR and Normal Equations, compute the average error over all the b_i.

Make two plots on a `semilogy` scale:

- the average error versus k for both QR and the Normal Equations,
- the condition number of A_k versus k.

Now tell me what's going on. More specifically:

1. What is the relationship between the error using QR versus the Normal Equations?

[1] `numpy.linalg.lstsq` in Python, `linsolve` in Matlab

2. What is the relationship between the errors and the condition number of A_k?

3. Suppose your matrix A is ill-conditioned. Which method is more favorable?

BONUS (10 POINTS): Take $k_{\max}$ up to 100. Something should break. What broke and why did it break? Any fixes?

BONUS (10 POINTS): Repeat this exercise (define the A_k's, etc.) with some other tall matrix you find in the wild. There are lots of examples from data science. What are the results? Why was the matrix you chose interesting? Any origin stories for the matrix (like, insight from the data that generated it) for why the condition number behaves the way it does?